



## **Time Complexity of Color Camera Depth Map Hand Edge Closing Recognition Algorithm**

Shahan Yamin Siddiqui<sup>1,2</sup>, Ghanwa Batool<sup>1</sup>, Muhammad Sohail Irshad<sup>1</sup>, Hafiz Muhammad Usama<sup>1</sup>, Muhammad Tariq Siddique<sup>1</sup>, Bilal Shoaib<sup>1</sup>, Sajid Farooq<sup>2</sup>, Arfa Hassan<sup>3</sup>

<sup>1</sup>Department of Computer Science, Minhaj University Lahore, Lahore, Pakistan.

<sup>2</sup>School of Computer Science, National College of Business Administration & Economics, Lahore, Pakistan.

<sup>3</sup>STS, University of Management & Technology, Lahore, Pakistan.

<sup>1,2</sup> [shahan.csit@mul.edu.pk](mailto:shahan.csit@mul.edu.pk), [sukainamustafvi@gmail.com](mailto:sukainamustafvi@gmail.com), [sohail.csit@mul.edu.pk](mailto:sohail.csit@mul.edu.pk), [usamatcs2137@gmail.com](mailto:usamatcs2137@gmail.com),  
<sup>1</sup>[tariqsiddique97@gmail.com](mailto:tariqsiddique97@gmail.com), <sup>1</sup>[bilal.shoaib@mul.edu.pk](mailto:bilal.shoaib@mul.edu.pk), <sup>2</sup>[sajidfarooq149@gmail.com](mailto:sajidfarooq149@gmail.com), <sup>3</sup>[arfach711@gmail.com](mailto:arfach711@gmail.com)

### **Abstract**

The objective of this paper is to calculate the time complexity of the colored camera depth map hand edge closing algorithm of the hand gesture recognition technique. It has been identified as hand gesture recognition through human-computer interaction using color camera and depth map technique, which is used to find the time complexity of the algorithms using 2D minima methods, brute force, and plane sweep. Human-computer interaction is a very much essential component of most people's daily life. The goal of gesture recognition research is to establish a system that can classify specific human gestures and can make its use to convey information for the device control. These methods have different input types and different classifiers and techniques to identify hand gestures. This paper includes the algorithm of one of the hand gesture recognition "Color camera depth map hand edge recognition" algorithm and its time complexity and simulation on MATLAB.

**Keywords:** Depth map, 2D minima, time complexity, MATLAB

### **1. INTRODUCTION**

Humans interact with each other mostly through speech, but some non-verbal means of communication are also used for interaction between humans [1]. The objective of creating a hand gesture

recognition technology is to generate a natural interaction between humans and computers. Where the recognized gestures can be used for monitoring a robot or can convey useful information [2]. Hand gestures are the source of delivering information that

is not easy to perceive. This research shows the different methodologies to detect and read the language of hand gestures for effective communication. Hand gesture recognition through robust marker-less hand segmentation method, Depth Map-based recognition Captured by colored Camera [3-4], Depth-Sensing Cameras based recognition and EMG Monitoring [5], using the convolutional neural network (CNN) to recognize gestures, based on Wireless Sensor, using Webcam, using a real-time tracking technique [6] and Markov hidden models, recognition through the machine is an important and efficient topic in human interaction system [7]. Human hand recognition is used in many areas such as image processing, cybersecurity, and robotics, etc. Hand gesture recognition is a very active research topic nowadays. Looking at the worth of gesture detection, experiments have been conducted to verify the results of application and system. In this article, we will find out the time complexity of color camera using a depth map-based method. The number of operations and computations are involved to compute time and task completion of a program/algorithm [8]. Time complexity is a more efficient term of operations in a program or system.

## **2. LITERATURE REVIEW**

Hand gesture recognition can be recognized by the human through its movements. But recognition of the human gesture by a machine is a big challenge. Some of the previous studies have been discussed in this article.

### ***2.1 Hand Gesture Recognition using Webcam***

In this article, it has been focused on creating a new way of communication or interaction between the computer and human and described a system that is

supposed to receive human gestures [9] as an input to control computer applications. The writer in this article has introduced a method that uses webcam through which gestures [10] provided by the user are captured, processed and the functions associated with that specific gestures are brought.

Our system introduced the use of python programming language for the movements of the cursor by hand gestures are done using OpenCV library, and the modules of python pyautogui [11-13] & NumPy [14-15]. The video captured will be fragmented into nonstop pictures edges on the behalf of functions defined in OpenCV for the recognition of gestures done by the handler [16]. In this paper for the detection process, they eliminate the mouse using a webcam. In future, it will be a matter of attraction as it doesn't want any physical contact with the device [17].

### ***2.2 A Simple and Effective Method for Hand Gesture Recognition***

In this paper, an efficient and simple scheme is presented. Usage of the skin color and labeling algorithm is introduced for hand recognition, it will do segmentation on the hand area from the background [18]. Then the center of mass & the palm point is used for the production of the baseline [19]. Shapes of hand gesture using the signature will be constructed [20]. As this can be forecast the labels of class will be classified in the last phase.

In this experiment original set of data [21] of 240 hand images are used for an experiment that is captured by a normal camera under the same condition. The main focus is towards the background, that background of the image should be very clean. The skin color is calculated by the HSV color model. The resultant output of the hand recognition is in the binary display. But the overall

performance of the proposed method in this article depends on the proper detection of hand gestures.

### ***2.3 Hand Gesture Recognition for Human-Computer Interaction***

Both the static and dynamic approaches are used for the detection of hand gestures [22]. In this paper, the gesture is detected for opening website, launching application like VLC, MS PowerPoint during working on computer. The interface between user & computer is the main area of this paper [23]. To obtain hand gesture recognition the two terminologies are used: Non-vision and vision-based. For some time, place and positioning of hand space does not change in static hand gesture, while, the waving of the hand is defined in a dynamic hand gesture.

Generally, this paper contains two parts, backend & frontend. There are three components of the backend described as the module of the camera, a module of detection and component of the interface [23]. Connecting and making a contribution through various sorts of picture identifiers and send pictures to the discovery module for further process as the working for edges has been done in the first segment [24-25]. The next component is responsible for image processing, it removes noise, a color conversion which is done by contour extraction [26]. The component in the last is responsible for the mapping of the hand gesture detected to their linked actions.

Frontend contains three layouts. Initial one gathers the video input captured by the camera with the relating name of the motion identified. The design demonstrates the shapes found in the picture [27]. The last format demonstrates the smooth threshold rendition of the picture.

RGB color space image is taken from the input image and then is cropped and it converts into a grayscale image. At that

point, the thresholding strategy is actualized to get a twofold picture from dark scale picture [28]. Shapes are a significant apparatus for item identification and acknowledgment in picture preparing, in this paper the technique to identify and perceive the hand from the foundation is displayed. Firstly, find a white object from the black background, besides drawing the forms which can be utilized to draw any shape gave the limit focuses. Further, they aimed to enhance and need more exactness to do the augmentation of actualizing capacities.

### ***2.4 Hand Gesture Recognition using a Real-Time Tracking Method and Hidden Markov Models***

The procedure has used to perceive consistent motion before a stationary foundation. This method recognizes the unknown input gesture with the use of HMMs. The writer has observed recent and used methods associated with hand gesture techniques [29], this technique is characterized into two methods glove based method and vision-based method [30-31]. This framework has four modules which are: an ongoing hand following and extraction, include extraction, concealed Markov display (HMM) preparation, and motion acknowledgment. The initial step is to apply on ongoing hand following and extraction calculation to follow the moving hand and concentrate the hand locale; at that point, we utilize the Fourier descriptor to depict spatial highlights and the movement examination to portray the worldly highlights. At that point, join the spatial and worldly highlights of the info picture arrange as our element vector. After removing the component vectors, we apply HMMs to perceive the info signal of hand. The signal to be perceived is independently recorded against various HMMs. In this article, the feature extraction method consists of different modules. Skin color detection will detect skin easily using

color information. And will detect the hand region by detecting skin regions from the image. The third module is edge detection; it is applied to the image to separate the arm area from the hand area. The model with the most elevated score demonstrates the proportional signal. This paper is not quite the same as other HMM-based signal acknowledgment frameworks. This framework does not utilize any instrumented glove, not any markers for acknowledgment, but rather, it utilizes 2D video contribution to the HMM-based motion acknowledgment framework in the examinations [32-33].

### ***2.5 A Real-time Hand Gesture Recognition and Human-Computer Interaction System***

In gestures hands are agile. It has three parts hand location, motion acknowledgment and human-interaction point [34]. They used CNN modified from LeNet-5 [35]. To recognized gesture and used a cheap monocular camera [36]. The monocular camera doesn't provide in-depth information such as separation of background from hands. Series of sequential and static 2D images can be captured using this cheap camera [37]. A natural communication network among people is hand gesture. So, HCI has application scenarios that are based on gestures [38]. Wearable electromagnetic devices and computer vision are two methods to perform gesture recognition [39]. The first one performs good, but the side-effect is it is costly and unusable. Features that are extracted by image processing are used for the performance of gesture recognition [40]. They have a set of gestures that can control mouse courser and it keeps a continuous trackable hand and to limit the mouse courser's movement, they use Kalman filter that causes smoothness and stability [41]. Serious of transient and intermediate gesture are caused by gestures that change from one to another. Hand gesture variability can lead

to a significant change in hand's shape that need a special point on hand and we don't know where the center of hand exists. By changing gestures, the hand's center is also changed. Palm's center would be a different form of fist's center.

### ***2.6 Hand Gesture Recognition Using Rgb-D Camera, Depth Map-Based***

A depth map & RGB Kinect camera, which give two types of info "Depth Map" and "RGB Image" [42]. They had used Depth map info to examine & diagnose the hand gesture. The proposed method consist of edge detection is also helpful to remove the noise and segment the hand [43]. This methodology is applied to recognize the French language sign alphabets to demonstrate its effectiveness & evaluation of the strength of the strategic descriptors is used [44]. The depth sensor is used for a better result; it consists of two methodologies: 1) Static & 2) Dynamic. Using static gestures along with the individual movements and image as an input. 23-static letters of the French alphabet are recognized by them [45]. The main purpose is to attain fast and precise hand gesture recognition [46] on depth map, which is conducted by Kinect camera.

The segmentation is an important module to detect the hands and their appearances in the image. Hand segmentation is comprised of 3 stages Edge detection, Edge closing, filling the Hand region, and removing all the unwanted edge regions. For the color image and the depth map, the Kinect sensor, which is an input device is used. The depth info is the main factor that gives each pixel depth for the sensor, firstly the depth info is converted into a greyscale image. To get a good result from the camera, the range will be in 1.2 to 3.5-meter distance and beyond this the accuracy of the sensor decreases rapidly.

There are 20 points which are detected in the human body. So, only the 12-point matched to the center of the right-hand palm from the 20 key point coordinates matrix. In hand segmentation, they remove all noise & useless info from the image by using the canny method [47]. It is a most effective method because it uses different threshold like strong & weak edges, by applying a bilateral filter which would preserve the edge and reduce the noise & then it will deliver to the canny method for more efficient work. It uses two types of structures first one consists of a 2-D structure that denotes the deformation of the hand in the 2-D strategy & demonstrates the geometry info & distinguishes the shapes of the hand structure.

### 3. SYSTEM MODEL

Our method has the following sequence, perform image acquisition, extract features from the image, localize the hand, recognize the hand and then will perform segmentation which has been presented in Figure 1.

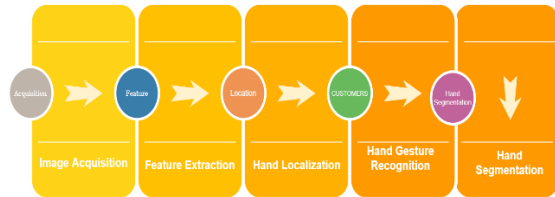


Figure 1: The system model of color camera hand gesture recognition method

### 4. COMPLEXITY ANALYSIS OF COLOR CAMERA DEPTH MAP HAND EDGE CLOSING RECOGNITION ALGORITHM

In this article, we will be calculating the time complexity of the color camera based on a depth map using 2D minima brute force and 2D minima plain sweep.

Table 1. Complexity Analysis of 2D Minima Brute Force Algorithm

And the second feature signifies the depth info 3-D mainly selected to signify the finger position and palm closure, they are based on the proposed recognition system. With the change of gesture, the hand position changed. The point of view is related to who established different hand gestures affectionate the alphabet of the American sign language [47].

Computational Intelligence approaches like Fuzzy system [48-52], Neural Network [53-57], Swarm Intelligence & Evolutionary Computing [58-62] like Genetic Algorithm, DE, Island GA, Island DE [49-58] are strong candidate solution in the field of the smart city [48], wireless communication [62], etc.

1	Brute force Minima ( $n, \text{Point}(Q1, \dots, QN)$ )		
2	For $K: 1$ to $n$		
3	Do Minimal $< - \text{time}$		
4	For $L: 1$ to $n$ If $((K \pm L) \text{ AND } [Q[K].x > Q[L].x] \text{ AND } [Q[K].y > Q[L].y])$ Then Minimal = false If (Minimal == true) Then Data $Q[L]$	$N$ $T1(n) = \Sigma(4)$ $L = 1$ $= 4n$	$N$ $T(n) = \Sigma(4n + 2)$ $K = 1$ $= 4n2 + 2n$ $= O(n2)$

Table 2. 2D Minima Plane Sweep Algorithm

1	Plane Sweep Minima ( $n, Q(1, \dots, N)$ )		
2	Set $Q$ in Increasing order by $x$ ; Stack $M$ ;		
3	For $K: 1$ to $n$  Do while ( $M.\text{notempty}() \&\& M.\text{top}().y > Q[K].y$ ) Do $M.\text{pop}()$ ; $M.\text{push}(Q[K])$ ; Output content of Stack $M$ ;	$T1(n) = n$	$T(n) = n + n \log 2n$ OR $T(n) = O(n \log 2n)$

Table 3. Closure of hand Regions Verification by Brute Force Minima

1	<i>ImgEdge</i> ; <i>minPixels</i> ; [ <i>objtEdge</i> ] = <i>FindObject</i> ( <i>ImgEdge</i> );		
2	for <i>k</i> = 1 to <i>Size(objtEdge)</i> [ <i>distObjt</i> ] = <i>distEuclidean</i> ( <i>minPixels</i> , <i>objtEdge</i> [ <i>i</i> ])	$T1(n, m, k)$ = $k * n * m$	$THRB(n, m, k) =$ $n * m[k + (n * m)]$ $+ c$
3	[ <i>h</i> ] = <i>Brute force Minima</i> ( <i>distObjt</i> ); If ( <i>Size(h)</i> == 1) [ <i>ImgTemp</i> ] ← [ <i>h</i> ]; [ <i>ImgTemp</i> ] ← <i>Thins</i> ( <i>ImgTemp</i> ); [ <i>endPoint</i> ] ← <i>FindEndpoints</i> ( <i>ImgTemp</i> ); if ( <i>Size(endPoint)</i> == 0) Return 1; Else Return 0;	$T2(n, m)$ = $(n * m) 2$ $+ c$	

Table 4. Closure of hand Regions Verification by 2D Minima Plane Sweep Algorithm

1	<i>ImgEdge</i> ; <i>minPixels</i> ; [ <i>objtEdge</i> ] = <i>FindObject</i> ( <i>ImgEdge</i> );	$T1(n, m, k) = k * n$ $* m$	
2	for <i>k</i> = 1 to <i>Size(objtEdge)</i> [ <i>distObjt</i> ] = <i>distEuclidean</i> ( <i>minPixels</i> , <i>objtEdge</i> [ <i>i</i> ])		$THRP(n, m, k)$ = $(k * n * m) + n$ $* m \log 2 n * m + c$ = $n * m[k + \log 2 n$ $* m]$ $+ c$
3	[ <i>h</i> ] = <i>Plane Sweep Minima</i> ( <i>distObjt</i> ); If ( <i>Size(h)</i> == 1) [ <i>ImgTemp</i> ] ← [ <i>h</i> ]; [ <i>ImgTemp</i> ] ← <i>Thins</i> ( <i>ImgTemp</i> ); [ <i>endPoint</i> ] = <i>FindEndpoints</i> ( <i>ImgTemp</i> ); if ( <i>Size(endPoint)</i> == 0) Return 1; Else Return 0;	$T2(n, m) = n * m \log 2 n * m + c$	

Table 5. Finding region of Circle

1	<i>ImgEdge</i> ; <i>circle_Array</i> ; [ <i>objtEdge</i> ] = <i>FindObject</i> ( <i>ImgEdge</i> );		
2	for <i>k</i> = 1 to <i>Size(objtEdge)</i> [[ <i>ImgTemp</i> ] = [ <i>objtEdge</i> [ <i>k</i> ]; [ <i>ImgTemp</i> ] = <i>Thins</i> ( <i>ImgTemp</i> ); [ <i>endPoints</i> ] = <i>FindEndpoints</i> ( <i>ImgTemp</i> ); if ( <i>Size(endPoint)</i> == 0) [ <i>circleObjt</i> ] = <i>Findcircle</i> ( <i>objtEdge</i> [ <i>k</i> ]); if ( <i>size</i> ( <i>circleObjt</i> )! = 0) [ <i>circle_Array</i> ] = <i>circleObjt</i> ; [ <i>ImgEdge</i> ] = <i>ImgEdge</i> - <i>ImgTemp</i> ;	$T1(n, m, k) = k * n$ $* m$	$TRC(n, m, k) = (k$ $* n$ $* m)$ $+ c$
3	Return ( <i>ImgEdge</i> , <i>circle_Array</i> );		

Table 6. Edge closure Verification

1	<i>ImgEdge</i> : <i>Bool</i> ;	
2	[ <i>ImTemp</i> ] ← [ <i>ImEdge</i> ]; [ <i>ImTemp</i> ] ← <i>Thins</i> ( <i>ImTemp</i> ); [ <i>endPoints</i> ] ← <i>FindEndpoints</i> ( <i>ImTemp</i> );	$TEC(n, m) = n * m$ $+ c$
3	If ( <i>Size(endPoints)</i> == 0) Return (1);	
4	Return (0);	

Table 7. Hand Edge Closing by Brute Force Minima

1	<i>ImgEdge</i> ; <i>ImgDepth</i> ;		
2	<i>minDepth</i> = <i>BruteForceMinima</i> ( <i>ImgDepth</i> ); [ <i>minPixels</i> ] = <i>Find</i> ( <i>minDepth</i> );	$TBF(n, m)$ = $(n * m) 2 + c$	
3	<i>r_Boolean</i> = <i>VerificationEdgeClosure</i> ( <i>ImgEdge</i> ); if ( <i>r_Boolean</i> == 0) [ <i>CircleArray</i> ] = <i>FindCircleArea</i> ( <i>ImgEdge</i> ); <i>h_Boolean</i> = <i>HandClosurebyBF</i> ( <i>ImgEdge</i> , <i>minPixels</i> ); if ( <i>h_Boolean</i> == 0) [ <i>ImgEdge</i> ] = <i>Dilate</i> ( <i>ImgEdge</i> ); [ <i>ImgEdge</i> ] = <i>Erode</i> ( <i>ImgEdge</i> );	$TEC(n, m) = n$ $* m$ $+ c$ $TRC(n, m, k)$ = $(k * n * m) + c$ $THRB(n, m, k) =$ $n * m[k + (n$ $* m)] + c$	$T(n, m, k)$ = $(n * m)[(n$ $* m) + 1 + k$ $+ k + (n * m)]$ $+ 4c$ = $(n * m)$ $[2(n * m) + 1$ $+ 2k] + 4c$
4	<i>ImgEdge</i> = <i>ImgEdge</i> + <i>CircleArray</i> ; Return ( <i>ImgEdge</i> );		

Table 8. Hand Edge Closing by Plane Sweep Minima

1	<i>ImgEdge</i> ; <i>ImgDepth</i> ;		
2	<i>minDepth</i> = <i>PlaneSweepMinima</i> ( <i>ImgDepth</i> ); [ <i>minPixels</i> ] = <i>Find</i> ( <i>minDepth</i> );	$TBF(n, m)$ = $n * m \log 2 n$ $* m + c$	
3	<i>r_Boolean</i> = <i>VerificationEdgeClosure</i> ( <i>ImgEdge</i> ); if ( <i>r_Boolean</i> == 0) [ <i>CircleArray</i> ] = <i>FindCircleArea</i> ( <i>ImgEdge</i> ); <i>h_Boolean</i> = <i>HandClosurebyPS</i> ( <i>ImgEdge</i> , <i>minPixels</i> ); if ( <i>h_Boolean</i> == 0) [ <i>ImgEdge</i> ] = <i>Dilate</i> ( <i>ImgEdge</i> ); [ <i>ImgEdge</i> ] = <i>Erode</i> ( <i>ImgEdge</i> );	$TEC(n, m) = n$ $* m$ $+ c$ $TRC(n, m, k)$ = $(k * n * m) + c$ $THRP(n, m, k) =$ $n * m[k + \log 2 n$ $* m] + c$	$(n * m)[\log 2 (n$ $* m) + 1 + k$ $+ k + \log 2 n$ $* m]$ $+ 4c$ = $(n$ $* m)[2 \log 2 (n$ $* m) + 1 + 2k]$ $+ 4c$
4	<i>ImgEdge</i> = <i>ImgEdge</i> + <i>CircleArray</i> ; Return ( <i>ImgEdge</i> );		

The Time Complexity (TC) of the Hand Gesture Edge Closing- (HGEC) by Brute Force minima algorithm and plane sweep minima algorithm which is presented in Table 9.

Table 9. Total Complexity

S#	Algorithm	Time Complexity
1	Hand Edge Closing by Brute Force Minima	$(n * m) [2(n * m) + 1 + 2k] + 4c$
2	Hand Edge Closing by Plane Sweep Minima	$(n * m) [2 \log 2 (n * m) + 1 + 2k]$ $+ 4c$

## 5. SIMULATION RESULTS AND DISCUSSION

Mat is used for simulation purposes. It is a multi-paradigm numerical computing environment & exclusive programming language established by Math Works. It has been observed in the simulation that if the image size ( $n \times m$ ) is increased the time complexity is also amplified. The 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> topmost curves represent the image size (500,300,200) simultaneously with several samples from 9:512 for 1<sup>st</sup> iteration and constant  $c=10$  for all iterations in HECBFM respectively which are presented in Figure 2.

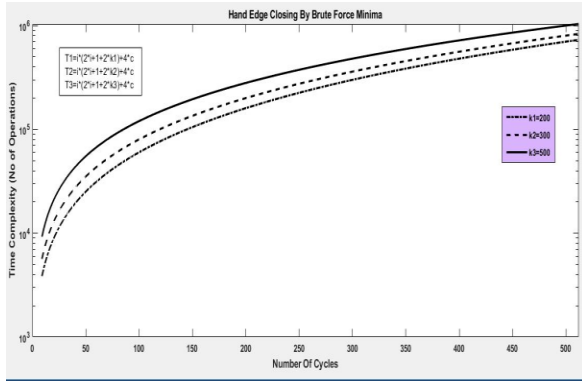


Figure 2: Time Complexity of HECBFM w.r.t Number of Cycles VS Size of Image

It has been observed in the simulation that if the size of the image ( $n \times m$ ) is increased the time complexity is also amplified. The 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> topmost curves represent the image size (500,300,200) simultaneously with several samples from 2:402 for 2<sup>nd</sup> iteration and constant  $c=10$  for all iterations in HECBFM respectively which are presented in Figure 3.

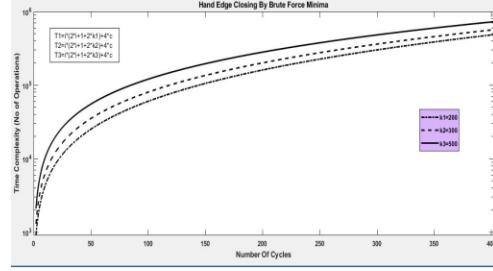


Figure 3: Time Complexity of HECBFM w.r.t Number of Cycles VS Size of Image

It has been observed in the simulation that if the size of the image ( $n \times m$ ) is increased the time complexity is also amplified. The 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> topmost curves represent the image size (500,300,200) simultaneously with several samples from 4:160 for 3<sup>rd</sup> iteration and constant  $c=10$  for all iterations in HECBFM respectively which are presented in Figure 4.

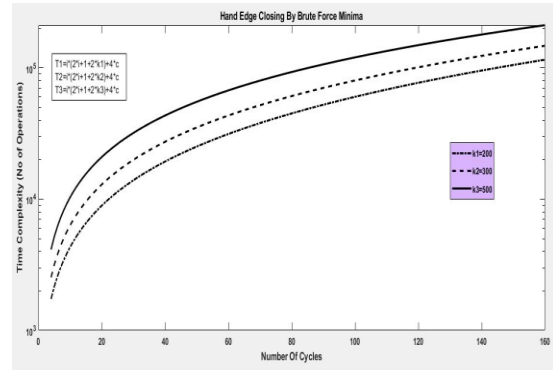


Figure 4: Time Complexity of HECBFM w.r.t Number of Cycles VS Size of Image

The TC of HEC by Brute Force Minima with respect to the number of cycles with different sizes of the image. It has been observed in the simulation that if the size of the image ( $n \times m$ ) is increased the time complexity is also amplified. The 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> topmost curves represent the image size (500,300,200) simultaneously with several samples from 6:560 for 4<sup>th</sup> iteration and constant  $c=10$  for all iterations in HECBFM respectively which are presented in Figure 5.

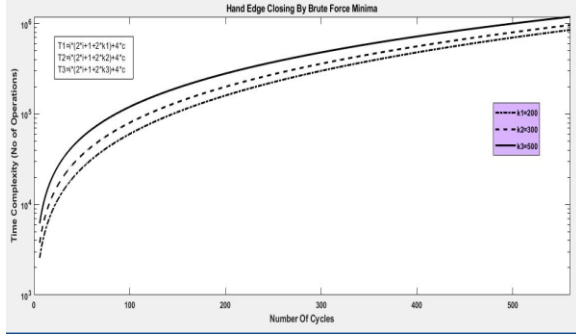


Figure 5: Time Complexity of HECBFM w.r.t Number of Cycles VS Size of Image

The TC of HEC by Plane Sweep Minima with respect to the number of cycles with different sizes of the image. It has been observed in the simulation that if the size of the image ( $n*m$ ) is increased the time complexity is also amplified. The 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> topmost curves represent the image size (500,300,200) simultaneously with several samples from 9:512 for 1<sup>st</sup> iteration and constant  $c=10$  for all iterations in HECPSM respectively which are presented in Figure 6.

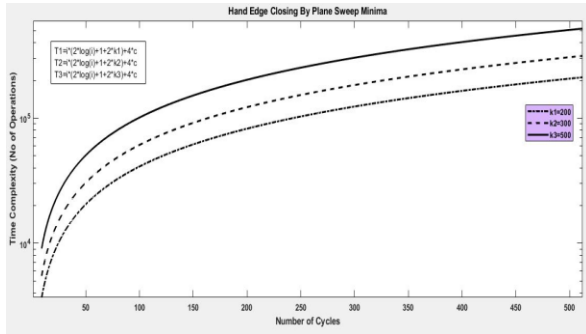


Figure 6: Time Complexity of HECPSM w.r.t Number of Cycles VS Size of Image

The TC of HEC by Plane Sweep Minima with respect to the number of cycles with different sizes of the image. It has been observed in the simulation that if the size of the image ( $n*m$ ) is increased the time complexity is also amplified. The 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> topmost curves represent the image size (500,300,200) simultaneously with several samples from 2:402 for 2<sup>nd</sup> iteration and constant  $c=10$  for

all iterations in HECPSM respectively which are presented in Figure 7.

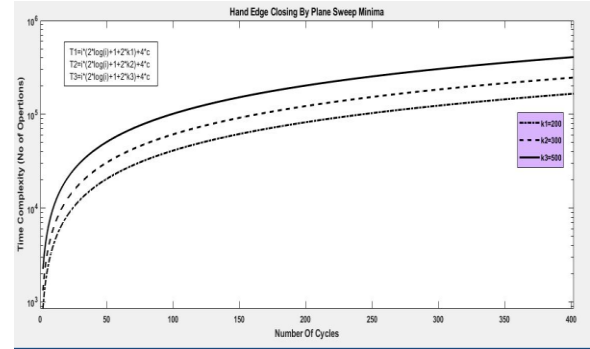


Figure 7: Time Complexity of HECPSM w.r.t Number of Cycles VS Size of Image

Figure 8 shows that the time complexity of Hand Edge Closing by Plane Sweep Minima w.r.t number of cycles with different sizes of the image. It has been observed in the simulation that if the size of the image ( $n*m$ ) is increased the time complexity is also amplified. The 1<sup>st</sup>, 2<sup>nd</sup>, and 3<sup>rd</sup> topmost curves represent the image size (500,300,200) simultaneously with several samples from 4:160 for 3<sup>rd</sup> iteration and constant  $c=10$  for all iterations in HECPSM respectively which are presented in Figure 8.

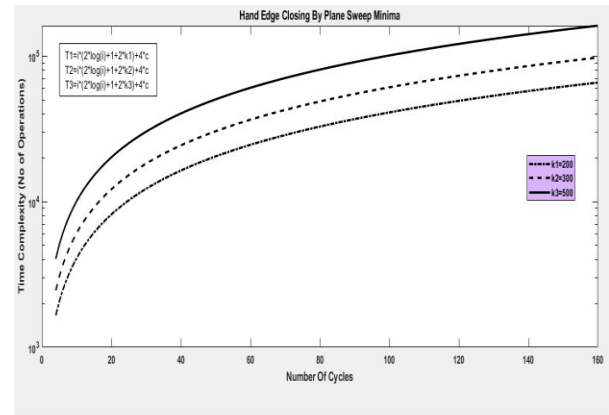


Figure 8: Time Complexity of HECPSM w.r.t Number of Cycles VS Size of Image

Figure 9 shows that the time complexity of Hand Edge Closing by Plane Sweep Minima

w.r.t number of cycles with different sizes of the image. It has been observed in the simulation that if the size of the image ( $n \times m$ ) is increased the time complexity is also amplified. The 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> topmost curves represent the image size (500,300,200) simultaneously with several samples from 6:560 for 4th iteration and constant  $c=10$  for all iterations in HECPSM respectively which are presented in Figure 9.

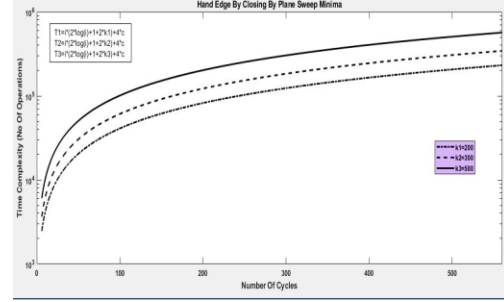


Figure 9: Time Complexity of HECPSM w.r.t Number of Cycles VS Size of Image

## 6. CONCLUSION

Hand gestures are the source of delivering information that is not easy to perceive. The scope of this area is vast these days in the IoT world as research shows the different methodologies to detect and read the sign language for effective communication. Hand gesture recognition is performed using robust marker-less hand gesture recognition system segmentation method and depth Map-based recognition using RGB-D Camera.

In this study, we had calculated the time complexity of the color camera depth map hand recognition method using 2D minima brute force and 2D minima plane sweep methodologies. The best approach for the hand closing edge algorithm is plane sweep as shown in Table 9.

## Reference

- [1] H. B. T. S. H. H. a. P. P. Z. Zafrulla, "American sign language recognition with the Kinect," in *ACM 13th international conference on multimodal interfaces*, 2011.
- [2] W. M. Y. B. J. A. B. H. Ahmed Ben Jmaa, "A New Approach For Hand Gestures Recognition Based on Depth Map Captured by RGB-D Camera," Vols. volume 20, N0 4, 2016.
- [3] W. C. L. C. Chen WL, "Depth-based hand gesture recognition using hand movements and defects," in *Next-Generation Electronics (ISNE)2015 International Symposium on*, Taiwan, Taipei; New York: IEEE;2015, 2015 May 4.
- [4] E. & M. T. M. Ohn-Bar, "Hand gesture recognition in real-time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Transactions on Intelligent Transportation Systems Vol. 15*, p. 2368–2377.
- [5] F. J. Ondrej Kainz, "Approach to Hand Tracking and Gesture Recognition Based on Depth-Sensing Cameras and EMG Monitoring," *Acata Informatica Pragensia*, vol. 3, no. 1, pp.104-112, 2014.
- [6] J. Y. J. M. a. Z. Z. Z. Ren, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Trans. Multimedia vol. 15, no. 5*, p. 1110–1120, 2013.
- [7] Itseez, "Open Source Computer Vision Library - itseez2015opencv," 01/2/2017.
- [8] "Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/Time\\_complexity](https://en.wikipedia.org/wiki/Time_complexity).
- [9] S. V. R. N. Ann Abraham Babu, "Hand Gesture Recognition System for Human-Computer Interaction Using Contour Analysis," *International Journal of Research in Engineering and Technology*, eISSN, pp. eISSN: 2319-1163, ISSN: 2321-7308.
- [10] A. N. Ming-Hsuan Yang, "Tabb, Extraction of 2D motion trajectories and its application to hand gesture recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on (Volume:24, Issue: 8)*, p. 2002, 1061 - 1074.
- [11] L. P. K. K. Madhuri, "Cursor Movements Controlled By Real-Time Hand Gestures," *International Journal of Science and Research (IJSR)*, India Online ISSN, pp. 2319-7064.
- [12] "Itseez," The OpenCV Reference Manual - itseez2014theopencv 2.4.9.0, April 2014.
- [13] "Python Software Foundation (PSF)," [Online]. Available: <https://pypi.python.org/pypi/PyAutoGUI>.
- [14] A. Sweigart, PyAutoGUI Documentation, 21/4/2017. [Online]. Available: <https://media.readthedocs.org/pdf/pyautogui/latest/pyautogui.pdf>.
- [15] S. C. C. a. G. V. Stéfan van der Walt, "The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering 13," no. 10.1109/MCSE.2011.37, 2011.
- [16] A. C.-B. a. G. M. Mark Bayazit, "Real-time motion-based gesture recognition using the GPU," in *IAPR Conference on Machine Vision Applications (MVA)*, 2009.
- [17] D. R. D. R. C. K. A. R. D. Athiya Marium\*, "Hand Gesture Recognition using Webcam".
- [18] J. L. Chunying QUAN, "A Simple and Effective Method for Hand Gesture Recognition".
- [19] H. B. T. S. H. H. a. P. P. Z. Zafrulla, "American sign language recognition with the Kinect," in *13th international conference on multimodal interfaces. ACM, 2011*.
- [20] H. P. a. J.-I. P. J. Choi, "Hand shape recognition using distance transform and

- shape decomposition," in *Image Processing (ICIP), 2011 18th IEEE International Conference*.
- [21] P. P. K. T. I. Dulayatrakul J, "Robust implementation of hand gesture recognition for remote human-machine interaction.," in *Robust implementation of hand gesture Information Technology and Electrical Engineering (ICITEE)*, 2015 Oct 29.
- [22] W. M. Y. B. J. A. B. H. Ahmed Ben Jmaa, "A New Approach For Hand Gestures Recognition Based on Depth Map Captured by RGB-D Camera," *ISSN*, 2007.
- [23] A. S. N. A. S. P., J. S. N. Aashni Hariaa, "Hand Gesture Recognition for Human-Computer Interaction," in *7th International Conference on Advances in Computing & Communications*, Cochi, India, 22-24 August 2011.
- [24] E. L. V. P. Granit Luzhnica, "A Sliding Window Approach to Natural Hand Gesture Recognition using a Custom Data Glove," in *3D User Interfaces (3DUI); 2016 IEEE Symposium on; 2016 Mar 19; New York: IEEE; 2016*.
- [25] Z. L. a. S. C. C. C. Wang, "Superpixel-Based Hand Gesture Recognition with Kinect Depth Camera," in *IEEE Transactions on Multimedia*, 2015.
- [26] H. K. Lee DH, "A Hand gesture recognition system based on difference image entropy," in *In: Advanced Information Management and Service (IMS), 2010 6th International Conference on; 2010 Nov 30, Seoul, New York: IEEE; 2010*.
- [27] Z. L. a. S. C. C. C. Wang, "A new high-resolution depth map estimation system using stereo vision and Kinect depth sensing," in *Journal of Signal Processing Systems* 79 (1).
- [28] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics* 1979; 9(1), pp. 62-66.
- [29] R. S. T. H. V.I. Pavlovic, "Visual interpretation of hand gestures for human-computer interaction, A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (7), p. 677-695, 1997.
- [30] M. B.-L. T. Baudel, "Charade: remote control of objects using free-hand gestures," *Comm. ACM* 36 (7), p. 28-35, 1993.
- [31] D. Z. D.J. Sturman, "A survey of glove-based input," *IEEE Computer Graphics and Applications* 14, p. 30-39, 1994.
- [32] D. B. A. A. A. B. A. P. L.W. Campbell, "Invariant features for 3-D gesture recognition," 1996.
- [33] E. H. R. J. J. Schlenzig, "Recursive identification of gesture inputs using hidden Markov models," in *Proceedings Second Annual Conference On Applications. Of Computer Vision*, 1994.
- [34] P. Xu\*, "A Real-time Hand Gesture Recognition and Human-Computer Interaction System," in *Department of Electrical and Computer Engineering*, University of Minnesota, Twin Cities.
- [35] A. C. a. J. L. Raheja, "A Health Monitoring System for Elder and Sick Persons," *Computer Theory and Engineering*, Vol. 5, No. 3, June 2013.
- [36] S. Pumrin\*, "Hand Gesture Recognition Algorithm for Smart cities based on wireless sensors," *International Journal of Online Engineering (iJOE)*, 2017.
- [37] D. K. a. E. A. Croft, "Safe planning for human-robot interaction. Journal of Field Robotics," pp. 383-396., 2005, 22(7).
- [38] S. H. G. a. C. C. Farcas, "Mouse Cursor Control System Based on Hand Gesture," *Procedia Technology*, pp. 657-661, 2016.
- [39] A. A. A. a. M. I. A. Lourakis, "Vision-based interpretation of hand gestures for remote control of a computer mouse," in *European Conference on Computer Vision. Springer Berlin Heidelberg*, 2006.
- [40] S. H. G. a. C. C. Farcas, "Mouse Cursor Control System Based on Hand Gesture. Procedia Technology," pp. 657-661, 2016, 22.
- [41] R. M. a. B. P. S. Thakur, " Vision-based computer mouse control using hand gestures.," in *Soft Computing Techniques and Implementations (ICSCTI), 2015 International Conference*, IEEE, 2015.
- [42] W. C. L. C. Chen WL, "Depth-based hand gesture recognition using hand movements and defects," in *In: Next-Generation Electronics (ISNE); 2015 International Symposium on; 2015 May 4; Taiwan, Taipei; New York: IEEE; 2015*.
- [43] J. Canny, "A computational approach to edge detection.," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, p. 679-698, 1986.
- [44] W. M. Y. B. J. A. B. H. Ahmed Ben Jmaa, "A New Approach For Hand Gestures Recognition Based on Depth Map Captured by RGB-D Camera," *Computations by Sistemas*.

- [45] S. & C. K. N. Ghosh, " On fast bilateral filtering using Fourier kernels," *IEEE Signal Processing Letters*, Vol. 23, p. pp. 570–573, 2016.
- [46] T. T. a. Y. W. C. Shan, ", Real-time hand tracking using a mean shift embedded particle filter," *Pattern Recognition*, vol. 40, pp. pp. 1958-1970, 2007.
- [47] F. C. J. & C. J. L. Berard, " Finger tracking as an input device for augmented reality," in Proc. Intel Workshop on Automatic Face and Gesture-Recognition, Zurich, Switzerland, 1995.
- [48] F. Areej, (2018). Intelligent Transportation System (ITS) for Smart-cities using Mamdani Fuzzy Inference System, *International Journal of Advanced Computer Science and Applications* (IJACSA). ISSN: 2158-107X, Vol. 9, No. 2, (pp. 94-105), Digital Object Identifier (DOI): 10.14569/IJACSA.2018.090215.
- [49] Muhammad Asadullah, Muhammad Adnan Khan, Sagheer Abbas, Atifa Athar, Syed Saqib Raza, and Gulzar Ahmad. Blind channel and data estimation using fuzzy logic-empowered opposite learning-based mutant particle swarm optimization. *Computational intelligence and neuroscience*, 2018, 2018.
- [50] M., Khan, M. A., & Choudry, M. A. S. (2013, January). GA backing to STBC based MC-CDMA systems. In 2013 4th International Conference on Intelligent Systems, Modelling, and Simulation (pp. 503-506). IEEE.
- [51] M. A., Umair, M., & Choudhry, M. A. S. (2015). GA based adaptive receiver for MC-CDMA system. *Turkish Journal of Electrical Engineering & Computer Sciences*, 23(Sup. 1), 2267-2277.
- [52] M. A., Umair, M., & Choudry, M. A. S. (2015, December). Island differential evolution based adaptive receiver for MC-CDMA system. In 2015 International Conference on Information and Communication Technologies (ICICT) (pp. 1-6). IEEE.
- [53] Ayesha Atta, Sagheer Abbas, M Adnan Khan, Gulzar Ahmed, and Umer Farooq. An adaptive approach: Smart traffic congestion control system. *Journal of King Saud University-Computer and Information Sciences*, 2018.
- [54] Umair, M., Khan, M. A., & Choudry, M. A. S. (2015, December). Island genetic algorithm-based MUD for MC-CDMA system. In 2015 International Conference on Information and Communication Technologies (ICICT) (pp. 1-6). IEEE.
- [55] M. N., Khan, M. A., Adeel, M., & Amir, M. (2016). Genetic Algorithm based adaptive Receiver for MC-CDMA system with variation in Mutation Operator. *International Journal of Computer Science and Information Security*, 14(9), 296.
- [56] S., Abbas, S., Sohail, T., Adnan Khan, M., & Athar, A. Estimating virtual trust of cognitive agents using multi-layered socio-fuzzy inference system. *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1-16.
- [57] M. A., Umair, M., Saleem, M. A., Ali, M. N., & Abbas, S. (2019). CDE using improved opposite-based swarm optimization for MIMO systems. *Journal of Intelligent & Fuzzy Systems*, 37(1), 687-692.
- [58] A. Fatima, Muhammad Adnan Khan, Sagheer Abbas, Muhammad Waqas, Leena Anum, and Muhammad Asif, Evaluation of Planet Factors of Smart City through Multi-layer Fuzzy Logic (MFL). *The ISC Int'l Journal of Information Security*. 51-58, 2019.
- [59] M., Khan, M. A., Abbas, S., & Saleem, M. (2019, January). Analysis of Space & Time Complexity with PSO Based Synchronous MC-CDMA System. In 2019 2nd international conference on computing, mathematics and engineering technologies (iCoMET) (pp. 1-5). IEEE.
- [60] Z. B., Khan, M. A., Abbas, S., & Fatima, A. (2019, January). Complexity analysis of DE based CE-UD MIMO system. In 2019 2nd international conference on computing, mathematics and engineering technologies (iCoMET) (pp. 1-5). IEEE.
- [61] M., Khan, M. A., Abbas, S., Alyas, T., Saleem, M. A., & Fatima, A. (2020). Blind Channel and Data Estimation Using Fuzzy Logic Empowered Cognitive and Social Information-Based PSO. *International Journal of Computational Intelligence Systems*, 22(1), 1-9.
- [62]

S. Abbas et al., "Modeling, Simulation and Optimization of Power Plant Energy Sustainability for IoT Enabled Smart Cities

Empowered With Deep Extreme Learning Machine," in IEEE Access, vol. 8, pp. 39982-39997, 2020, doi: 10.1109/ACCESS.2020.2976452.